UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/601,575	06/24/2003	Martin Robert Evans	550-445	8224

23117 7590 05/31/2006

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
LAI, VINCENT	
ART UNIT	**PAPER NUMBER**
2181	

DATE MAILED: 05/31/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/601,575	EVANS ET AL.	
	Examiner	Art Unit	
	Vincent Lai	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *06 April 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Fritz Fleming
Supervisory FRITZ FLEMING
PRIMARY EXAMINER 5/26/2006
GROUP 2100
Au 2181

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Acknowledgment is made of the amendment of the specification filed by applicant on 6 April 2006

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 1/8/2004 was considered by the examiner.

Drawings

3. The drawings are objected to because element 230 in Figure 2A is mislabeled as element 23 (The office is assuming the label for the element is incorrect or else the specification is incorrect). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary

to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: "Synchronisation Between Pipelines in a Data Processing Apparatus Utilizing Synchronisation Queues".

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

5. Claims 1-45 are rejected under 35 U.S.C. 102(b) as being anticipated by Brown, III et al (U.S. Patent # 6,240,508, B1.

As per claim 1, Brown, III et al discloses a data processing apparatus,

comprising:

a main processor operable to execute a sequence of instructions (Column 35, lines 45-49), the main processor comprising a first pipeline having a first plurality of pipeline stages (Column 12, lines 18-24);

a coprocessor operable to execute coprocessor instructions in said sequence of instructions (Column 35, lines 45-49), the coprocessor comprising a second pipeline having a second plurality of pipeline stages, and each coprocessor instruction being arranged to be routed through both the first pipeline and the second pipeline (Column 36, lines 21-22, and column 36, line 67- column 37, lines 3: There is one instruction stream but instructions are issued to only one processor); and

at least one synchronising queue coupling a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines (Column 49, lines 58-60), the predetermined pipeline stage being operable to cause a token to be placed in the synchronising queue when processing a coprocessor instruction and the partner pipeline stage being operable to process that coprocessor instruction upon receipt of the token (Column 50, lines 6-11: Token is the commands associated with the spec queue) from the synchronising queue, thereby synchronising the first and second pipelines between the predetermined pipeline stage and the partner pipeline stage (Column 35, lines 43-44: Synchronization is done with multiple pipelined processors).

As per claim 2, Brown, III et al discloses a plurality of said synchronising queues, each said synchronising queue coupling a predetermined pipeline stage in one of the

pipelines with a partner pipeline stage in the other of the pipelines (Column 49, lines 60-63).

As per claim 3, Brown, III et al discloses wherein one of the at least one synchronising queues is an instruction queue (Column 10, lines 4-10), the predetermined pipeline stage is in the first pipeline and is arranged to cause a token identifying a coprocessor instruction to be placed in the instruction queue, and the partner pipeline stage is in the second pipeline and is operable upon receipt of the token to begin processing the coprocessor instruction identified by the token (Column 35, lines 61-63: The second processor must wait until processing can be done).

As per claim 4, Brown, III et al discloses wherein the predetermined pipeline stage is a fetch stage in the first pipeline and the partner pipeline stage is a decode stage in the second pipeline, that decode stage being operable to decode the coprocessor instruction upon receipt of the token (Column 12, line 64- column 13, line 1: Decomposing data is the same as decoding data.).

As per claim 5, Brown, III et al discloses wherein the fetch stage in the first pipeline is operable to cause a token to be placed in the instruction queue for each instruction in the sequence of instructions (Column 49, lines 58-63), and the decode stage in the second pipeline is arranged to decode each instruction upon receipt of the associated token in order to determine whether that instruction is a coprocessor

instruction that requires further processing by the coprocessor (Column 49, lines 63-65: Decode is part of the I-box).

As per claim 6, Brown, III et al discloses wherein one of the at least one synchronising queues is a cancel queue (Column 49, lines 63-65: The stop-spec-queue acts as a cancel), the predetermined pipeline stage is in the first pipeline and is arranged to cause to be placed in the cancel queue a token identifying whether a coprocessor instruction at that predetermined pipeline stage is to be cancelled (Column 49, lines 60-63), and the partner pipeline stage is in the second pipeline and is operable upon receipt of the token from the cancel queue (Column 36, lines 40-42), and if the token identifies that the coprocessor instruction is to be cancelled, to cause that coprocessor instruction to be cancelled (Column 36, lines 42-55: Coprocessor will not be able to process its instruction in the case of a invalidate).

As per claim 7, Brown, III et al discloses wherein the predetermined pipeline stage is an issue stage in the first pipeline, and the partner pipeline stage is a stage following an issue stage in the second pipeline (Column 13, lines 19-26: Issue stage is followed by decoding of issued instruction).

As per claim 8, Brown, III et al discloses wherein the partner pipeline stage is operable upon receipt of the token from the cancel queue (Column 36, lines 40-42), and

if the token identifies that the coprocessor instruction is to be cancelled, to remove the coprocessor instruction from the second pipeline (Column 36, lines 42-55).

As per claim 9, Brown, III et al discloses wherein one of the at least one synchronising queues is a finish queue (Column 50, lines 24-31), the predetermined pipeline stage is in the first pipeline and is arranged to cause to be placed in the finish queue a token identifying permission for a coprocessor instruction at that predetermined pipeline stage to be retired from the second pipeline (Column 49, lines 60-63), and the partner pipeline stage is in the second pipeline and is operable upon receipt of the token from the finish queue, and if the token identifies that the coprocessor instruction is permitted to be retired, to cause that coprocessor instruction to be retired (Column 50, lines 24-31).

As per claim 10, Brown, III et al discloses wherein the predetermined pipeline stage is a write back stage in the first pipeline, and the partner pipeline stage is a write back stage in the second pipeline (Column 48, lines 41-45).

As per claim 11, Brown, III et al discloses wherein one of the at least one synchronising queues is a length queue (Column 28, lines 50-53), the predetermined pipeline stage is in the second pipeline and is arranged, for a vectored coprocessor instruction, to cause to be placed in the length queue a token identifying length information for the vectored coprocessor instruction (Column 49, lines 60-63), and the

partner pipeline stage is in the first pipeline and is operable upon receipt of the token from the length queue to factor the length information into the further processing of the vectored coprocessor instruction within the first pipeline (Column 28, line 50-53).

As per claim 12, Brown, III et al discloses wherein the predetermined pipeline stage is a decode stage in the second pipeline, and the partner pipeline stage is a first execute stage in the first pipeline (Column 13, lines 19-26: Instructions are decoded and then send to the execute stage).

As per claim 13, Brown, III et al discloses wherein one of the at least one synchronising queues is an accept queue (Column 35, lines 33-36), the predetermined pipeline stage is in the second pipeline and is arranged to cause to be placed in the accept queue a token identifying whether a coprocessor instruction in that predetermined pipeline stage is to be accepted for execution by the coprocessor (Column 49, lines 60-63), and the partner pipeline stage is in the first pipeline and is operable upon receipt of the token from the accept queue, and if the token identifies that the coprocessor instruction is not to be accepted, to cause that coprocessor instruction to be rejected by the main processor (Column 35, lines 33-36).

As per claim 14, Brown, III et al discloses wherein the predetermined pipeline stage is an issue stage in the second pipeline, and the partner pipeline stage is a second execute stage in the first pipeline (Column 13, lines 19-26).

As per claim 15, Brown, III et al discloses wherein the partner pipeline stage is operable upon receipt of the token from the accept queue, and if the token identifies that the coprocessor instruction is not to be accepted, to remove the coprocessor instruction from the first pipeline (Column 50, lines 19-23)

As per claim 16, Brown, III et al discloses wherein one of the at least one synchronising queues is a store queue used when the coprocessor instruction is a store instruction operable to cause data items to be transferred from the coprocessor to memory accessible by the main processor (Column 48, lines 41-45), the predetermined pipeline stage is in the second pipeline and is arranged, when processing one of said store instructions, to cause to be placed in the store queue a token identifying each data item to be transferred (Column 49, lines 60-63), and the partner pipeline stage is in the first pipeline and is operable upon receipt of each token from the store queue, to cause the corresponding data item to be transferred to the memory (Column 48, lines 41-45).

As per claim 17, Brown, III et al discloses wherein the predetermined pipeline stage is an issue stage in the second pipeline, and the partner pipeline stage is an address generation stage in the first pipeline (Column 12, line 64- column 13, line 1).

As per claim 18, Brown, III et al discloses wherein one of the at least one synchronising queues is a load queue used when the coprocessor instruction is a load

instruction operable to cause data items to be transferred from memory accessible by the main processor to the coprocessor (Column 12, line 64- column 13, line 1: Done by prefetching operands), the predetermined pipeline stage is in the first pipeline and is arranged, when processing one of said load instructions, to cause to be placed in the load queue a token identifying each data item to be transferred (Column 49, lines 60-63), and the partner pipeline stage is in the second pipeline and is operable upon receipt of each token from the load queue, to cause the corresponding data item to be transferred to the coprocessor (Column 12, line 64- column 13, line 1).

As per claim 19, Brown, III et al discloses wherein the predetermined pipeline stage is a write back stage in the first pipeline, and the partner pipeline stage is a write back stage in the second pipeline (Column 48, lines 41-45).

As per claim 20, Brown, III et al discloses wherein one of the at least one synchronising queues is a store queue used when the coprocessor instruction is a store instruction operable to cause data items to be transferred from the coprocessor to memory accessible by the main processor (Column 48, lines 41-45), the predetermined pipeline stage is in the second pipeline and is arranged, when processing one of said store instructions, to cause to be placed in the store queue a token identifying each data item to be transferred (Column 49, lines 60-63), and the partner pipeline stage is in the first pipeline and is operable upon receipt of each token from the store queue, to cause the corresponding data item to be transferred to the memory, and wherein the load

instruction and store instruction may be vectored coprocessor instructions defining multiple data items to be transferred (Column 48, lines 41-45), and the apparatus further comprises flow control logic, associated with at least one of the load queue and the store queue, operable to send a control signal to the predetermined pipeline stage to stop issuance of tokens by the predetermined pipeline stage whilst it is determined that the associated load or store queue may become full (Column 25, lines 43-47).

As per claim 21, Brown, III et al discloses wherein the flow control logic is provided for the store queue, the flow control logic being operable to issue the control signal upon receiving an indication from the main processor that the partner pipeline stage cannot accept a data item (Column 41, lines 51-54).

As per claim 22, Brown, III et al discloses wherein the load queue is a double buffer (Column 7, line 62- column 8, line 1: Data is prefetched in quadwords).

As per claim 23, Brown, III et al discloses wherein each token includes a tag which identifies the coprocessor instruction to which the token relates (Column 45, lines 12-14).

As per claim 24, Brown, III et al discloses wherein the main processor is operable, when it is necessary to flush coprocessor instructions from both the first and the second pipeline, to broadcast a flush signal to the coprocessor identifying the tag

relating to the oldest instruction that needs to be flushed, the coprocessor being operable to identify that oldest instruction from the tag and to flush from the second pipeline that oldest instruction and any later instructions within the coprocessor (Column 41, lines 20-25).

As per claim 25, Brown, III et al discloses wherein one or more of said at least one synchronising queues are flushed in response to said flush signal, with the tag being used to identify which tokens within the queue are to be flushed (Column 41, lines 21-27).

As per claim 26, Brown, III et al discloses wherein the at least one synchronising queue comprises a First-In-First-Out (FIFO) buffer having a predetermined number of entries for storing tokens (Column 28, lines 50-53).

As per claim 27, Brown, III et al discloses wherein a plurality of said coprocessors are provided, with each synchronising queue coupling a pipeline stage in the main processor with a pipeline stage in one of the coprocessors (Figure 18, element 75, and column 23, lines 12-17: The bus is connected to the spec queue, which helps synchronize the pipeline).

As per claim 28 Brown, III et al discloses wherein the data processing apparatus has a synchronous design (Column 9, lines 4-8), such that the tokens are caused to be

placed in the queue by the predetermined pipeline stage and are caused to be received from the queue by the partner pipeline stage upon changing edges of a clock cycle (Column 9, lines 38-57).

As per claim 29, Brown, III et al discloses a method of synchronisation between pipelines in a data processing apparatus, the data processing apparatus comprising a main processor operable to execute a sequence of instructions (Column 35, lines 45-49) and a coprocessor operable to execute coprocessor instructions in said sequence of instructions (Column 35, lines 45-49), the main processor comprising a first pipeline having a first plurality of pipeline stages (Column 12, lines 18-24), and the coprocessor comprising a second pipeline having a second plurality of pipeline stages, (Column 36, lines 21-22, and column 36, line 67- column 37, line 3) and each coprocessor instruction being arranged to be routed through both the first pipeline and the second pipeline, the method comprising the steps of:

(a) coupling a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines via a synchronising queue (Column 49, lines 58-60);

(b) placing a token in the synchronising queue when the predetermined pipeline stage is processing a coprocessor instruction (Column 50, lines 6-11);

(c) upon receipt of the token from the synchronising queue by the partner pipeline stage, processing the coprocessor instruction within the partner pipeline stage;

whereby synchronisation of the first and second pipelines between the predetermined pipeline stage and the partner pipeline stage is obtained (Column 35, lines 43-44).

As per claim 30, Brown, III et al discloses wherein a plurality of said synchronising queues are provided, and said steps (a) to (c) are performed for each synchronising queue (Column 49, lines 60-63).

As per claim 31, Brown, III et al discloses wherein one of the at least one synchronising queues is an instruction queue (Column 10, lines 4-10), the predetermined pipeline stage is in the first pipeline and the partner pipeline stage is in the second pipeline, the method comprising the steps of: at said step (b), placing a token in the instruction queue identifying a coprocessor instruction; and at said step (c), upon receipt of the token, beginning processing of the coprocessor instruction identified by the token within the partner pipeline stage (Column 35, lines 61-63).

As per claim 32, Brown, III et al discloses wherein one of the at least one synchronising queues is a cancel queue (Column 49, lines 63-65), the predetermined pipeline stage is in the first pipeline and the partner pipeline stage is in the second pipeline, the method comprising the steps of: at said step (b), placing a token in the cancel queue identifying whether a coprocessor instruction at that predetermined pipeline stage is to be cancelled (Column 49, lines 60-63); and at said step (c), upon receipt of the token from the cancel queue by the partner pipeline stage (Column 36,

lines 40-42), and if the token identifies that the coprocessor instruction is to be cancelled, causing that coprocessor instruction to be cancelled (Column 36, lines 42-55).

As per claim 33, Brown, III et al discloses wherein one of the at least one synchronising queues is a finish queue (Column 50, lines 24-31), the predetermined pipeline stage is in the first pipeline and the partner pipeline stage is in the second pipeline, the method comprising the steps of: at said step (b), placing in the finish queue a token identifying permission for a coprocessor instruction at that predetermined pipeline stage to be retired from the second pipeline (Column 49, lines 60-63); and at said step (c), upon receipt of the token from the finish queue by the partner pipeline stage, and if the token identifies that the coprocessor instruction is permitted to be retired, causing that coprocessor instruction to be retired (Column 50, lines 24-31)

As per claim 34, Brown, III et al discloses wherein one of the at least one synchronising queues is a length queue (Column 28, lines 50-53), the predetermined pipeline stage is in the second pipeline and the partner pipeline stage is in the first pipeline, and the method comprises the steps of: at said step (b), for a vectored coprocessor instruction, placing in the length queue a token identifying length information for the vectored coprocessor instruction (Column 49, lines 60-63); and at said step (c), upon receipt of the token from the length queue by the partner pipeline

stage, factoring the length information into the further processing of the vectored coprocessor instruction within the first pipeline (Column 28, lines 50-53).

As per claim 35, Brown, III et al discloses wherein one of the at least one synchronising queues is an accept queue (Column 35, lines 33-36), the predetermined pipeline stage is in the second pipeline and the partner pipeline stage is in the first pipeline, the method comprising the steps of: at said step (b), placing in the accept queue a token identifying whether a coprocessor instruction in that predetermined pipeline stage is to be accepted for execution by the coprocessor (Column 49, lines 60-63); and at said step (c), upon receipt of the token from the accept queue by the partner pipeline stage, and if the token identifies that the coprocessor instruction is not to be accepted, causing that coprocessor instruction to be rejected by the main processor (Column 35, lines 33-36).

As per claim 36, Brown, III et al discloses wherein one of the at least one synchronising queues is a store queue used when the coprocessor instruction is a store instruction operable to cause data items to be transferred from the coprocessor to memory accessible by the main processor (Column 48, lines 41-45), the predetermined pipeline stage is in the second pipeline and the partner pipeline stage is in the first pipeline, the method comprising the steps of: at said step (b), when processing one of said store instructions, placing in the store queue a token identifying each data item to be transferred (Column 49, lines 60-63); and at said step (c), upon receipt of each token

from the store queue by the partner pipeline stage, causing the corresponding data item to be transferred to the memory (Column 48, lines 41-45).

As per claim 37, Brown, III et al discloses wherein one of the at least one synchronising queues is a load queue used when the coprocessor instruction is a load instruction operable to cause data items to be transferred from memory accessible by the main processor to the coprocessor (Column 12, line 64- column 13, line 1), the predetermined pipeline stage is in the first pipeline and the partner pipeline stage is in the second pipeline, the method comprising the steps of: at said step (b), when processing one of said load instructions, placing in the load queue a token identifying each data item to be transferred (Column 49, lines 60-63); and at said step (c), upon receipt of each token from the load queue by the partner pipeline stage, causing the corresponding data item to be transferred to the coprocessor (Column 12, line 64-column 13, line 1).

As per claim 38, Brown, III et al discloses wherein one of the at least one synchronising queues is a store queue used when the coprocessor instruction is a store instruction operable to cause data items to be transferred from the coprocessor to memory accessible by the main processor (Column 48, lines 41-45), the predetermined pipeline stage is in the second pipeline and the partner pipeline stage is in the first pipeline, the method comprising the steps of: at said step (b), when processing one of said store instructions, placing in the store queue a token identifying each data item to

be transferred (Column 49, lines 60-63); and at said step (c), upon receipt of each token from the store queue by the partner pipeline stage, causing the corresponding data item to be transferred to the memory; and wherein the load instruction and store instruction may be vectored coprocessor instructions defining multiple data items to be transferred (Column 48, lines 41-45), and the method further comprises the step of: (d) for at least one of the load queue and the store queue, sending a control signal to the predetermined pipeline stage to stop issuance of tokens by the predetermined pipeline stage whilst it is determined that the associated load or store queue may become full (Column 25, lines 43-47).

As per claim 39, Brown, III et al discloses wherein said step (d) is performed for the store queue, at said step (d) the method comprising the step of issuing the control signal upon receiving an indication from the main processor that the partner pipeline stage cannot accept a data item (Column 41, lines 51-54).

As per claim 40, Brown, III et al discloses wherein each token includes a tag which identifies the coprocessor instruction to which the token relates (Column 45, lines 12-14).

As per claim 41, Brown, III et al discloses wherein, when it is necessary to flush coprocessor instructions from both the first and the second pipeline, the method further comprises the steps of: broadcasting a flush signal from the main processor to the

coprocessor identifying the tag relating to the oldest instruction that needs to be flushed; within the coprocessor, identifying from the tag that oldest instruction and flushing from the second pipeline that oldest instruction and any later instructions within the coprocessor (Column 41, lines 20-25).

As per claim 42, Brown, III et al discloses further comprising the step of flushing one or more of said at least one synchronising queues in response to said flush signal, with the tag being used to identify which tokens within the queue are to be flushed (Column 41, liens 21-27).

As per claim 43, Brown, III et al discloses wherein the at least one synchronising queue comprises a First-In-First-Out (FIFO) buffer having a predetermined number of entries for storing tokens (Column 28, lines 50-53).

As per claim 44, Brown, III et al discloses wherein a plurality of said coprocessors are provided, with each synchronising queue coupling a pipeline stage in the main processor with a pipeline stage in one of the coprocessors (Figure 18, element 75, and column 23, lines 12-17).

As per claim 45, Brown, III et al discloses wherein the data processing apparatus has a synchronous design (Column 9, lines 4-8), such that the tokens are placed in the

queue by the predetermined pipeline stage and are received from the queue by the partner pipeline stage upon changing edges of a clock cycle (Column 9, lines 38-57).

Response to Arguments

6. Regarding figure 2A, the examiner submits the figure that was originally examined, with the error circled. Examiner admits that included drawing with remarks to first office action does reflect no error was made; however, no record of new drawings have been made in between the time of the original submission and the submission of the remarks to the first office action. If applicant had submitted new drawings in the time period described above, the drawings must be resubmitted to be placed into the record of actions. Otherwise, the submission of the figure 2A with the remarks must be submitted according to normal office procedures.

7. It is noted that the applicant did not address the examiner's issue with the title of the invention.

8. Applicant's arguments filed on 6 April 2006 have been fully considered but they are not persuasive.

Applicant argues "CPU 28 is not described in Brown as a coprocessor for the CPU 10."

In column 35, lines 45-49, CPU 10 and CPU 28 are described as processor-1 and processor-2, respectively, and are stated to be a part of a multiprocessor environment. It is implied by a multiprocessor system that any other processors in the environment can be a coprocessor for the processor in question. Brown does disclose processors can work together (See column 36, line 62- column 37, line 3: Macropipeline considerations mean that instructions are split up among different processors) and thus CPU 10 and CPU 28 can be considered coprocessors and thus the argument is seen as non-persuasive.

Applicant argues "Brown's CPU 28 does not execute CPU 28 instructions appearing the same sequence of instructions executed by the CPU 10."

Examiner notes the exact wording "the same sequence of instructions" is not used in any of the claims, as implied by the underlining. However, examiner does recognize such implications are made in the claims. Given the evidence of macropipelining described above, this argument is seen as non-persuasive.

Applicant argues "Brown does not disclose that CPU 28 is pipelined."

Examiner would like to point to column 8, lines 23-26. It has been established CPU 10 is pipelined and CPU 28 is described as being able to have the same design as CPU 10, and thus CPU 28 will be pipelined. This argument is seen as non-persuasive.

Applicant argues instructions are not routed through both pipelines and that "the examiner admits that Brown's instructions are issued to only one processor."

Examiner would like to point out a misinterpretation. Instructions are issued to only one processor, but this does not mean that instructions are not routed to other processors. This means that one processor (being the main processor) in charge of routing instructions to the other processors (the coprocessors). In the macropipeline, instructions are routed to other processors. Given this interpretation, the argument is seen as non-persuasive.

Applicant argues "no such [synchronizing] queue is described in Brown."

By disclosing a sync-queue, Brown is implying its use with multiple pipelines. Synchronization implies two or more separate components are necessary. A synchronized queue in a single pipeline would essentially be a pipeline that requires all instructions to be executed to be in the order of issue. Thus Brown meant the spec-queue to be used in synchronizing the different pipelines of the multiprocessor system, which has been explained above. Given this interpretation, the argument is seen as non-persuasive.

Applicant argues "how are [the] commands placed in the synchronizing queue as tokens by the predetermined pipeline stage when processing a coprocessor instruction."

Synchronizing memory references is necessary as it is the main purpose of synchronizing an instruction. If a memory reference is incorrect then subsequent

instructions may be affected. A pipeline stage will request a memory reference to obtain instruction/data to manipulate. When such occurrence happens, a command/address is created. Brown defines the spec-queue as being able to hold commands and addresses. Since the multiprocessor system has one main memory system, it can be seen that these commands will affect each other. If microinstructions come from the same macroinstruction and a request is made for memory, the memory management unit will ensure that the proper order is kept so that data will be correct. This is how commands are placed in synchronizing queues as tokens by the predetermined pipeline stage when processing a coprocessor instruction. Given this interpretation, the argument is seen as non-persuasive.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz M. Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Vincent Lai
Examiner
Art Unit 2181

vl
May 24, 2006

Supervisory

FRITZ FLEMING
PRIMARY EXAMINER 5/26/2006
GROUP 2100
AU 2181

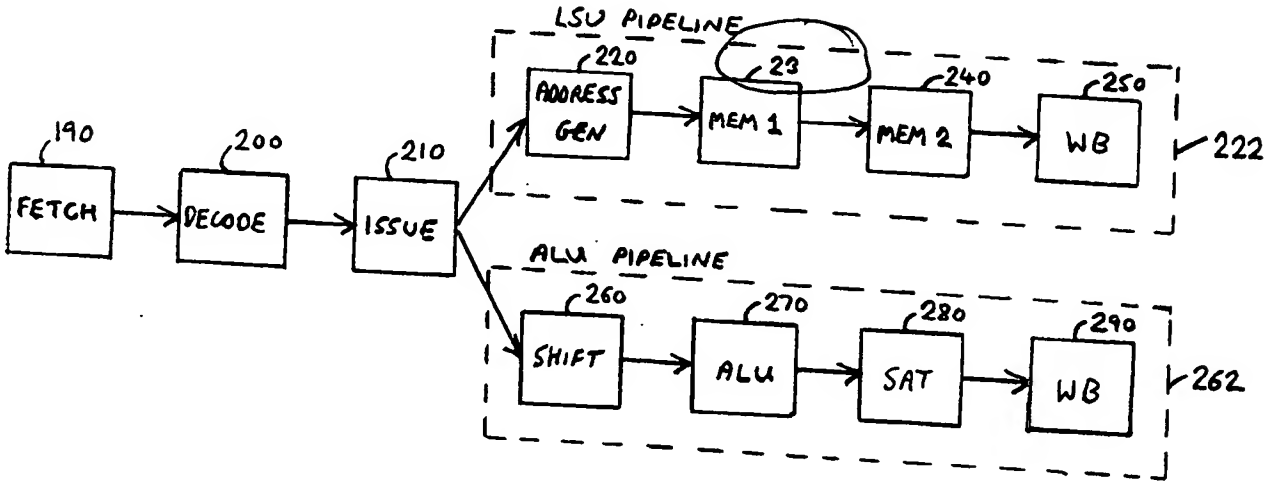
LSU PIPELINE
ALU PIPELINE
190
FETCH
200
DECODE
210
ISSUE
220
ADDRESS GEN
23
MEM 1
240
MEM 2
250
WB
222
260
SHIFT
270
ALU
280
SAT
290
WB
262

FIG 2A

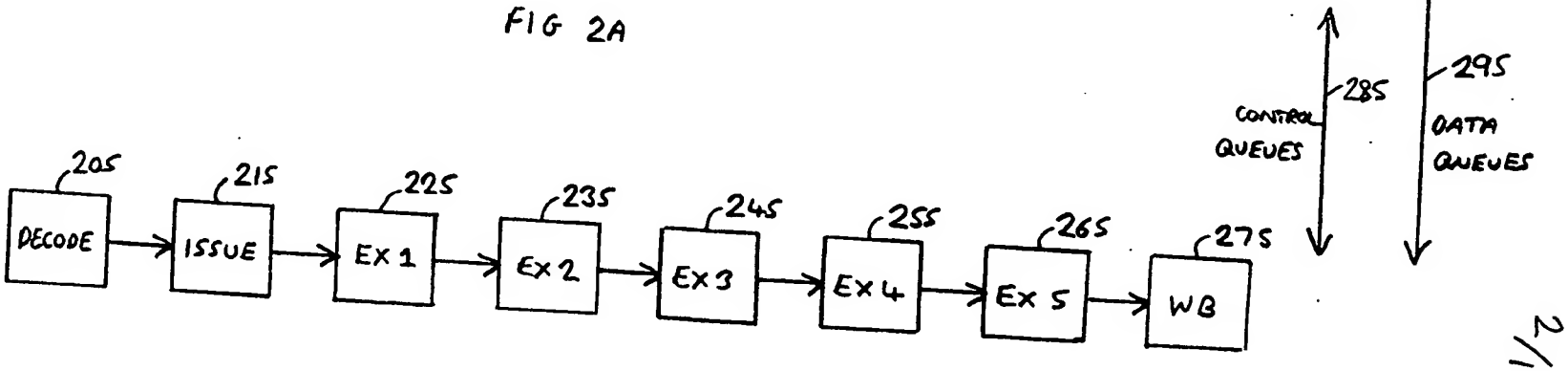
205
DECODE
215
ISSUE
225
EX 1
235
EX 2
245
EX 3
255
EX 4
265
EX 5
275
WB
285
CONTROL QUEUES
295
DATA QUEUES

FIG 2B